

Бармина Алиса БПИ208

**Важно:** при написании программы использовалась версия CMake 3.20 и стандарт /std:c++latest языка программирования C++ (нужно для использования binary\_semaphore и #include<semaphore>)

## Пояснительная записка

### Описание

Номер задания: 4

Пять философов сидят возле круглого стола. Они проводят жизнь, чередуя приемы пищи и размышления. В центре стола находится большое блюдо спагетти. Спагетти длинные и запутанные, философам тяжело управляться с ними, поэтому каждый из них, чтобы съесть порцию, должен пользоваться двумя вилками. К несчастью, философам дали только пять вилок. Между каждой парой философов лежит одна вилка, поэтому эти высококультурные и предельно вежливые люди договорились, что каждый будет пользоваться только теми вилками, которые лежат рядом с ним (слева и справа). Написать многопоточную программу, моделирующую поведение философов с помощью семафоров. Программа должна избегать фатальной ситуации, в которой все философы голодны, но ни один из них не может взять обе вилки (например, каждый из философов держит по одной вилки и не хочет отдавать ее). Решение должно быть симметричным, то есть все потоки-философы должны выполнять один и тот же код.

### Входные и выходные данные

Входные данные не требуются.

Выходные данные:

==> Start

Далее в течение 3 циклов выводятся данные следующего формата:

Philosopher#<n> is thinking

Philosopher#<n> takes his left fork (fork#<n>)

Philosopher#<n> takes his right fork (fork#<(n + 1) % 5>)

Philosopher#<n> has eaten

Демонстрируя поочередность процессов думы и трапезы, а также показывая очередность взятия вилок философами.

Пример вывода:

```
==> Start
Philosopher#2 is thinking
Philosopher#1 is thinking
Philosopher#3 is thinking
Philosopher#4 is thinking
Philosopher#5 is thinking
Philosopher#2 takes his left fork (fork #2)
Philosopher#2 takes his right fork (fork #3)
Philosopher#1 takes his left fork (fork #1)
Philosopher#4 takes his left fork (fork #4)
Philosopher#4 takes his right fork (fork #5)
Philosopher#2 has eaten
Philosopher#1 takes his right fork (fork #2)
Philosopher#3 takes his left fork (fork #3)
Philosopher#2 is thinking
Philosopher#4 has eaten
Philosopher#4 is thinking
Philosopher#3 takes his right fork (fork #4)
Philosopher#1 has eaten
Philosopher#1 is thinking
Philosopher#5 takes his left fork (fork #5)
Philosopher#2 takes his left fork (fork #2)
Philosopher#5 takes his right fork (fork #1)
Philosopher#3 has eaten
Philosopher#2 takes his right fork (fork #3)
Philosopher#4 takes his left fork (fork #4)
Philosopher#3 is thinking
Philosopher#5 has eaten
Philosopher#5 is thinking
Philosopher#4 takes his right fork (fork #5)
Philosopher#1 takes his left fork (fork #1)
Philosopher#2 has eaten
Philosopher#1 takes his right fork (fork #2)
Philosopher#3 takes his left fork (fork #3)
Philosopher#2 is thinking
Philosopher#4 has eaten
Philosopher#4 is thinking
Philosopher#3 takes his right fork (fork #4)
Philosopher#1 has eaten
Philosopher#5 takes his left fork (fork #5)
Philosopher#1 is thinking
Philosopher#5 takes his right fork (fork #1)
Philosopher#2 takes his left fork (fork #2)
Philosopher#3 has eaten
Philosopher#2 takes his right fork (fork #3)
Philosopher#4 takes his left fork (fork #4)
Philosopher#3 is thinking
Philosopher#5 has eaten
Philosopher#1 takes his left fork (fork #1)
Philosopher#5 is thinking
Philosopher#4 takes his right fork (fork #5)
Philosopher#2 has eaten
Philosopher#1 takes his right fork (fork #2)
Philosopher#3 takes his left fork (fork #3)
Philosopher#4 has eaten
Philosopher#3 takes his right fork (fork #4)
Philosopher#1 has eaten
Philosopher#5 takes his left fork (fork #5)
Philosopher#5 takes his right fork (fork #1)
Philosopher#3 has eaten
Philosopher#5 has eaten
==> Finish
```